

# Laohu cluster user manual

Li Changhua

National Astronomical Observatory,  
Chinese Academy of Sciences

2011/12/26

# About laohu cluster

- Laohu cluster has 85 hosts , each host has 8 CPUs and 2 GPUs. GPU is Nvidia C1060, and the CUDA lib version is 4.0, please ref. <http://laohu.bao.ac.cn>.
- Laohu cluster has a login node, `laohu.bao.ac.cn`. user can login by ssh. After login, you can upload and download files, compile code. (The SSH user guide references appendix I ).
- Laohu cluster has firewall, which deploy whitelisting system. Please send me your machine internet IP address before login.
- Special Host group information:  
host\_study (c1104 c0504 c0407 c1106 c1107), is scheduled by the cpu\_test and gpu\_test queue.

# About LSF

- **Platform LSF (Load Sharing Facility) is a suite of distributed resource management products that:**
  - Connects computers into a Cluster (or “Grid”) ;
  - Monitors loads of systems ;
  - Distributes, schedules and balances workload;
  - Controls access and load by policies;
  - Analyzes the workload;
  - High Performance Computing (HPC) environment;
- **LSF can schedules, controls, and tracks the job according to configured Policies, so, any job or task can be run on the best node available.**

the main command for job management: bqueues, bsub, bjobs, bkill, bpeek etc.

# Displays queues

Sample: **bqueues**

---

QUEUE_NAME	PRIO	STATUS	MAX	JL/U	JL/P	JL/H	NJOBS	PEND	RUN	SUSP
normal	30	Open : Active	-	8	-	-	22	2	20	0
long	30	Open : Active	-	304	-	-	52	12	40	0

---

Notes:

- **QUEUE\_NAME:** The name of the queue
- **PRIO:** The priority of the queue, The larger the value, the higher the priority.
- **STATUS:** The current status of the queue.
  - Open:Active The queue is able to accept jobs and Jobs in the queue may be started.
  - Closed:Active The queue is not able to accept jobs , but Jobs in the queue may be started.
- **MAX:** The maximum number of job slots that can be used by the jobs from the queue. “ - ” means no limit.
- **JL/U:** The maximum number of job slots each user can use for jobs in the queue.
- **NJOBS:** The total number of job slots held currently by jobs in the queue.
- **PEND:** The number of job slots used by pending jobs in the queue
- **RUN:** The number of job slots used by running jobs in the queue
- **SUSP:** The number of job slots used by suspended jobs in the queue.

# Displays queue detailed information

Sample: `bqueues -l`

---

```
QUEUE: normal
  -- For normal priority jobs , allow 2-8 CPU cores , Max Job Slots is 40 This is the default queue.

PARAMETERS/STATISTICS
PRIO NICE STATUS          MAX JL /U JL /P JL /H NJOBS  PEND  RUN  SSUSP  USUSP  RSV
 30   20  Open: Active      -   40   -   -   288   48   240    0    0    0

CPULIMIT                PROCLIMIT
345600.0 min of E5410   2 2 8

PROCESSLIMIT
8
```

---

Notes:

**CPULIMIT:** The maximum CPU time a job can use, in minutes, relative to the CPU factor of the named host.

**PROCLIMIT:** The [minimum],[default],maximum number of processors allocated to a job.

**PROCESSLIMIT:** The maximum number of concurrent processes allocated to a job.

# Current Queue Configuration

- `cpu_32`: the maximum number of job slot can allocate to a job is 192, the minimum number is 150.
- `cpu_large`: The maximum number of job slot can allocate to a job is 128, the minimum number is 30.
- `cpu_small`: The maximum number of job slot can allocate to a job is 48, the minimum number is 1.
- `cpu_test`: the maximum number of job slot is 12.
- `gpu_32`: the number of job slot is between 32 and 16.
- `gpu_16`: the number of job slot is between 16 and 8.
- `gpu_8`: the number of job slot is between 8 and 4.
- `gpu_test`: the number of job slot is between 10 and 1.

# Submits a job

Command format:

```
bsub [options] command [arguments]
```

- Options: command options, such as `-n,-q,-R` etc.
- Arguments: program arguments

After submitted, you can execute `bjobs` to query host running your jobs, then, you can ssh to this host and execute `top` command to display the cpu/mem usage rate of this host.

# Submits a job to a special queue

- Option `-q` specifies a special queue. Sample 1:

```
bsub -q serial executable1
```

submits job to queue serial.

If succeeds, displays output as follows:

---

```
Job <79722> is submitted to default queue <serial >
```

---

79722 is the job id, which is unique.

# Specifies the CPU number required

- Option `-n` specifies the number of processors required to run the job (some of the processors may be on the same multiprocessor host).

About GPU application, suggests apply 2 CPUs and 2 GPUs in a host, About CPU application, no GPU computing, suggests apply 6 CPUs in a host.

In order to avoid several CPU processes to use same GPU, please don't use a const gpu device id in your program.

# Other useful option of bsub

- Specify a standard output and error output file `-i -o -e`  
Sample 1: `bsub -i executable1.input -o executable1-%J.log  
-e executable1-%J.err executable1`

- Exclusive execution mode `-x`

In exclusive execution mode, your job runs by itself on a host. It is dispatched only to a host with no other jobs running, and LSF does not send any other jobs to the host until the job completes.

- Resource requirement `-R`

Sample 2: `bsub -R 'select[type==any] span[ptile=1]  
usage[ngpus=2]' executable1`

Runs the job on hosts that meets the specified resource requirements

# Submits Job by Script

Sample 1: `cpujob.lsf`

```
#!/bin/sh
```

```
#BSUB -q naoc_c #job queue, modify according to user
```

```
#BSUB -a openmpi-qlc
```

```
#BSUB -R 'select[type==any] span[ptile=6] '
```

```
#resource requirement of host
```

```
#BSUB -o out.test
```

```
#output file
```

```
#BSUB -n 256
```

```
#the maximum number of CPU
```

```
mpirun.lsf --mca "btl openib,self" Gadget2wy WJL.PARAM  
# need modify for user's program.
```

Exec method: **`bsub < cpujob.lsf`**

# Submits Job by Script

Sample 2: `gpujob.lsf`

```
#!/bin/sh
```

```
#BSUB -q naoc_g    #job queue
```

```
#BSUB -a openmpi-qlc
```

```
#BSUB -R 'select[type==any] span[ptile=2] rusage[ngpus=2]'
```

```
#resource requirement of host
```

```
#BSUB -o out.test
```

```
#output file
```

```
#BSUB -n 20
```

```
#the maximum number of CPU
```

```
mpirun.lsf --prefix "/usr/mpi/gcc/openmpi-1.3.2-qlc" -x
```

```
"LD_LIBRARY_PATH=/export/cuda/lib:/usr/mpi/gcc/openmpi-1.3.2-qlc/lib64" ./phi-GRAPE.exe
```

```
# need modify for user's program.
```

Exec method: `bsub < gpujob.lsf`

# Displays job status

## Sample 1: `bjobs`

---

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
79726	hmli	RUN	normal	user	2*node31 1*node4	*executab1	Mar 12 19:20
79727	hmli	PEND	long	user		*executab2	Mar 12 19:20

---

Job 79726 has 2 process in host node31 and 1 process in host node4. Job 79727 is pending.

## Sample 2: `bjobs -l 79727`

---

```
Job Id <79727>, User <hmli>, Project <default>, Status <PEND>,
Queue <long> , Command <executab2>
Sun Mar 12 14:15:07: Submitted from host <hpc1.ustc.edu.cn>,
CWD <$HOME>, Requested Resources <type==any && swp>35>;
```

```
PENDING REASONS:
```

```
The user has reached his/her job slot limit;
```

```
SCHEDULING PARAMETERS:
```

	r15s	rlm	r15m	ut	pg	io	ls	it	tmp	swp	mem
loadSched	-	0.7	1.0	-	4.0	-	-	-	-	-	-
loadStop	-	1.5	2.5	-	8.0	-	-	-	-	-	-

---

# Terminates a job

- Sample: **bkil 79722**  
Job <79722> is being terminated.
- Please terminates the job that has errors or don't run later, so as to free host resource.

# Suspends and resumes job

- Sample: `bstop 79727`

Notes:

1. Suspends unfinished jobs, so other pending job can start to run.

2. As to the job is running, even if it be suspended, but it don't free using resource. So, I suggest that you shouldn't suspend running jobs.

- Sample: `bresume 79727`

Continues to run job 79727.

# Modifies pending job

- Sample 1: *btop 79727*

Moves job 79727 to the top of queue.

- Sample 2: *bbot 79727*

Moves job 79727 to the bottom of queue.

- Sample 3:

*bmod -Z executable2 -q long -n 12 79727*

modifies job 79727 's executable program, queue and CPU number.

# Displays output of job

- Sample 1: `bpeek 79727`

Displays the stdout and stderr output of job 79727

- Sample 2: `bpeek -f 79727`

Displays the stdout and stderr output of job 79727 using the command `tail -f`

- If you use the parameter `-o`, `-e` in `bsub` command, you can open the specified file to display output.

# Displays Host Status

- Sample 1: **bhosts**

---

HOST_NAME	STATUS	JL/U	MAX	NJOBS	RUN	SSUSP	USUSP	RSV
node12	closed	-	4	2	2	0	0	0
node10	ok	-	2	2	1	0	0	0

---

STATUS列中的ok表示可以接收新作业，closed表示已经被占满。

- Sample 2: **lsload**

---

HOST_NAME	status	r15s	rlm	r15m	ut	pg	ls	it	tmp	swp	mem
node10	ok	0.0	0.0	0.0	0%	3.5	0	2050	9032M	4000M	16G
node11	locku	0.0	0.0	0.0	0%	3.5	0	2050	9032M	4000M	16G

---

ut列表示利用率，status列中的locku表示在进行排他性运行。

# Displays user's information

- Sample 1: **busers lich**

---

USER/GROUP	JL / P	MAX	NJOBS	PEND	RUN	SSUSP	USUSP	RSV
lich	-	22	40	32	8	0	0	0

---

- Notes:
  1. MAX: The maximum number of job slots that can be processed concurrently for user lich' jobs.
  2. NJOBS: The current number of job slots used by user lich' jobs.
  3. PEND: The number of pending job slots used by jobs of user lich.
  4. RUN: The number of job slots used by running jobs of user lich.

# Appendix I SSH Introduction

- Under linux :

```
ssh user@laohu.bao.ac.cn
```

when popup the password dialog window, please fill your correct password.

you can use scp command to copy file between your local machine and remote cluster login host.

```
scp localfile user@laohu.bao.ac.cn:~/
```

- Under windows:

First, installs a ssh client soft., such as SecureShellClient which is simple to use. About download address and user guide, please reference another documentation [seureshellclient.pdf](#)